

(19)  Canadian
Intellectual Property
Office

An Agency of
Industry Canada

Office de la Propriété
Intellectuelle
du Canada

Un organisme
d'Industrie Canada

(11) **CA 2 279 145** (13) **A1**

(43) 01.02.2000

(12)

(21) 2 279 145

(51) Int. Cl.⁸: **G06F 017/30, H04N 007/01**

(22) 30.07.1999

(30) 09/127,613 US 01.08.1998

(71) **COREL CORPORATION,**
1600 Carling Avenue, OTTAWA, O1 (CA).

(72) **CALL, JUSTIN D. (US).**

(74) **SIM & MCBURNEY**

(54) **HYPERLIEN A VISIONNEMENT PREALABLE DU DOCUMENT**
(54) **HYPERTEXT LINK PREVIEW**

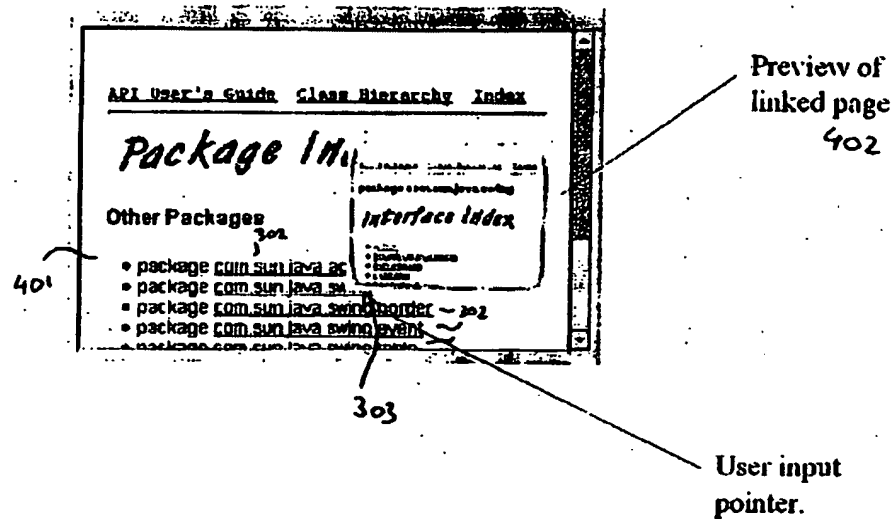
(57)

In a computer application for displaying
hypertext documents, a system and method for displaying
a preview of a linked document before the link is
activated by the user. The displayed preview is
generated from the linked document itself, in order to
provide an accurate and timely representation of the
linked document.



(21) (A1) **2,279,145**
(22) 1999/07/30
(43) 2000/02/01

(72) CALL, JUSTIN D., US
(71) COREL CORPORATION, CA
(51) Int.Cl.⁶ G06F 17/30, H04N 7/01
(30) 1998/08/01 (09/127,613) US
(54) **HYPERLIEN A VISIONNEMENT PREALABLE DU DOCUMENT**
(54) **HYPERTEXT LINK PREVIEW**



(57) In a computer application for displaying hypertext documents, a system and method for displaying a preview of a linked document before the link is activated by the user. The displayed preview is generated from the linked document itself, in order to provide an accurate and timely representation of the linked document.



HYPERTEXT LINK PREVIEW

ABSTRACT OF THE DISCLOSURE

In a computer application for displaying hypertext documents, a system and method for displaying a preview of a linked document before the link is activated by
5 the user. The displayed preview is generated from the linked document itself, in order to provide an accurate and timely representation of the linked document.

HYPertext LINK PREVIEW**Inventor:****Justin D. Call****BACKGROUND OF THE INVENTION**

5

1. Field of the Invention

The present invention relates generally to software applications such as browsers for viewing hypertext documents, and more particularly to a system and method of displaying a preview of a linked document before the link is activated.

10 **2. Description of Background Art**

Hypertext is a form of text, usually displayed on a computer screen, which facilitates linking from one document to another. In recent years, standardized formats for hypertext encoding have emerged, such as for example Hypertext Markup Language (HTML), Extended Markup Language (XML), Standardized
15 General Markup Language (SGML), and the like. In general, such hypertext languages provide codes for specifying links from one document to another. Links are usually associated with a particular word, phrase, graphic, or portion of a graphic, in a document. Each such link specifies a document that is to be retrieved when the link is activated. Many links, often leading to different documents, may be provided in a
20 single document.

Currently, the most common use for hypertext is as a medium for navigating the World Wide Web and similar computer networks. By specifying links on a document (or page), the document author is able to provide a mechanism for permitting the user to retrieve associated documents from various web sites, as well
25 as navigate among documents within a site.

Users view and navigate hypertext documents using software such as a browser. Several popular browsers are available, including for example, Netscape® Navigator™ from Netscape Communications, and Microsoft® Internet Explorer™ from Microsoft Corporation. Operation of such browsers is generally quite similar.

Documents encoded in HTML or a similar language are displayed on a computer screen as "pages", formatted according to specified codes embedded in the document, and often including graphical elements. Links to other documents may also be present on a displayed document, and the text associated with a link is often indicated by the use of a distinct font style and/or color. In some cases, a graphic element or a portion thereof may be associated with a link. In order to activate a link on a displayed document, the user typically moves a mouse or similar pointing device so as to move an on-screen cursor onto the text or graphic associated with the link, and clicks a button. This causes the document associated with the selected link to be retrieved.

In most conventional browsers, the user is not able to view a linked document until he or she activates the link. In many situations, the user may not be sure whether a particular link will retrieve a document that is of interest. Generally, hypertext documents provide some clues as to the nature of the linked document, giving the user some guidance in deciding which link to activate next. The particular text or graphic associated with a link is often the best source of such information.

However, the information provided in the first, or source, document itself is often insufficient in describing the linked document. Browsers sometimes provide features for improving the information available to the user before clicking on a link. For example, both Navigator and Internet Explorer momentarily display the file name of a linked document whenever the user positions the on-screen cursor over the link. However, in many situations the file name is obscure and/or meaningless, since it is essentially an arbitrary name selected by the document author.

Some browsers also provide functionality for displaying a "ToolTip" when the on-screen cursor is placed on top of a link. This ToolTip displays a short line of text, which presumably can help describe the link. However, the text displayed in the ToolTip is specified by the document author, as an ALT tag in the HTML code specifying the link parameters; if the author does not provide such a tag, or provides inadequate text for the tag, the ToolTip display may not be helpful to the user.

Some web pages include more sophisticated methods of describing links, using scripting environments such as JavaScript or similar. Using such techniques, a graphic and/or short description of a link may be presented to the user when the cursor is placed over a link. However, such techniques still employ information from the script presented in the source document, and do not use information from the linked document. Therefore the descriptions presented are not automatically generated or updated based on the content of the linked document. Furthermore, such techniques rely on the source document's author to provide the necessary programming and data to present the link previews. Thus, they are poor solutions for automated generation and update of usable and detailed link previews.

In general, all the above techniques rely on information in the source document, and do not display any information taken from the linked document itself. Where the source document does not provide adequate information, there is no way, using the above techniques, for the user to be presented with sufficient "preview" information to decide whether to follow a link. Furthermore, any information taken from the source document can potentially become outdated, as the linked document may change without the source document being updated accordingly.

What is needed is a system and method of displaying an up-to-date preview of a linked document that provides sufficient information regarding the linked document so that a user can decide whether or not to activate a link. What is further needed is a preview system and method that does not rely on a document author to provide information about a linked document. What is further needed is a preview system and method that obtains preview information from the linked document. Finally, what is further needed is a preview system and method that reflects updated information concerning the linked document, without manual update of the source document.

SUMMARY OF THE INVENTION

In accordance with the present invention, there is provided a system and method of displaying a preview of a linked document, in response to a user showing some interest in the document. For example, an image resembling either a miniature
5 version of the linked document, or a portion of the linked document, can be displayed when the user passes an on-screen cursor over the link. By providing a graphic representation of the linked document, the present invention is able to provide more information than conventional methods which display only a limited text-based description of a link that is based solely on information from the source
10 document.

In the present invention, the displayed preview image is based on the linked document itself, rather than on any hard-coded information from the source document. Thus, the displayed preview is up-to-date even if the source document has not been changed to reflect the changes in the linked document.

15 In one embodiment, preview data for a link is retrieved and cached in advance using a background process, so that previews can be displayed without undue delay when the user places the cursor over a link. Link preview resources for displaying previews for all links on a source document begin to load as soon as the source document has been loaded. When the user indicates interest in a particular link, the
20 corresponding preview can immediately be displayed if its resources have been loaded. If the loading has not yet been performed or completed when a user expresses interest in a link, the present invention initiates immediate loading of the resources needed for that link. When such loading is complete, the preview is displayed for the user. Display of the preview may be stopped when appropriate, for
25 example when a predefined period of time has elapsed, or when the user indicates that he or she is no longer interested in the link.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a system for practicing the present invention.

Fig. 2 is an example of a hypertext document containing links to other documents.

5 Fig. 3 is an example of a screen display showing a hypertext document according to the prior art.

Fig. 4 is an example of a screen display showing a hypertext document and a link preview according to the present invention.

Fig. 5 is a block diagram of the functional components of one embodiment of
10 the present invention.

Fig. 6A is a timeline diagram showing a page loading operation according to the present invention.

Fig. 6B is a flowchart showing a page loading operation according to the present invention.

15 Fig. 7A is a timeline diagram showing a link previewing operation according to the present invention.

Fig. 7B is a flowchart showing a link previewing operation according to the present invention.

Fig. 8A is a timeline diagram showing a preview stopping operation according
20 to the present invention.

Fig. 8B is a flowchart showing a preview stopping operation according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring now to Fig. 1, there is shown a block diagram for practicing the
25 present invention. In one embodiment, the present invention is implemented on a personal computer running the Microsoft® Windows™ 95 operating system on an Intel® Pentium® processor. The present invention may be implemented as a feature in a software application such as a browser running on the aforementioned hardware.

One skilled in the art will recognize that other embodiments, including other types of software applications, processors, and operating systems, are also possible.

In the block diagram of Fig. 1, computer 100 is shown having a central processing unit (CPU) 101, display device 102, input device such as a pointing device 5 103, random access memory (RAM) 104, and storage device 105. Storage device 105 also includes a cache 106 for temporary storage of link preview data, as will be described in more detail below. Also provided is a connection 107 to a network such as the Internet. The following detailed description of the invention will make reference to exemplary implementations of such components, though other embodiments may 10 also be used. For example, CPU 101 is a microprocessor such as an Intel Pentium processor; display device 102 is a conventional monitor or screen such as a cathode-ray tube (CRT); pointing device 103 is a mouse or trackball, though other input devices such as keyboards can also be used; RAM 104 is some quantity of conventional memory as is commonly supplied with personal computers; storage 15 device 105 is a hard disk or similar device for long-term storage of programs and data; cache 106 is implemented in some portion of storage device 105, though cache 106 can also be implemented in RAM 104 as is known in the art; and Internet connection 107 is implemented using known protocols such as Transmission Control Protocol / Internet Protocol (TCP/IP) across a modem, T1 or T3 line, or other 20 connection medium. Documents to be viewed and previewed using the present invention can be retrieved across Internet connection 107 using known techniques. Internet connection 107 is not necessary to practice the present invention, as the invention can also be implemented for viewing hypertext documents that are stored locally, such as on storage device 105, for example.

25 A user runs a software application by loading the software onto computer 100 (from storage device 105) and invoking a command to begin execution. Typically, the software application is then loaded into RAM 104 so that CPU 101 can have access to the program instructions contained therein. On some computer systems, certain

portions of the software application may remain on disk 105 to be "swapped into" RAM 104 only as needed, in order to conserve memory space.

One example of such a software application is a browser. Such browsers are well known in the art, including for example Netscape Navigator and Microsoft Internet Explorer. For illustrative purposes, the remainder of this description will focus on browser applications for viewing documents and pages on a network over an Internet connection, though the present invention can easily be implemented in connection with other types of applications and documents as well, such as for example word processing applications and/or locally-stored documents. For purposes of this description, the terms "document" and "page" are used interchangeably, though other types of documents or files could also be used without departing from the spirit or essential characteristics of the invention.

Referring now to Fig. 2, there is shown an example of a hypertext document 200. Document 200 is stored as an HTML file at some location on a remote server, or may be locally stored. As is known in the art, a user may retrieve document 200 using an Internet Protocol (IP) address or Uniform Resource Locator (URL) specifying the location of document 200, and document 200 is then displayed on a screen 102 using browser software.

Document 200 contains text and/or graphics, as shown in the example of Fig. 2. Links 201, 202, 203 to other documents 211, 212, 213 may be included within document 200, and may be represented as graphic elements 201, 202, or as text 203. Links 203 associated with text are typically displayed in a unique color, typeface, and/or style to indicate to the user that the associated text is a link. Links 201, 202 associated with graphic elements may be indicated by a colored border, if desired. Other means of highlighting links 201, 202, 203 may be employed, and such means may be selectable or configurable by the user. In some cases, it may be desirable not to highlight or indicate the presence of links.

For purposes of this application, a document such as document 200 is referred to as a "source" document, whereas those such as documents 211, 212, 213 are referred to as "linked" documents.

As is known in the art, a user may retrieve a linked document 211, 212, 213 by
5 activating the appropriate link 201, 202, 203 corresponding to the desired document. Thus, it is advantageous if link 201, 202, 203 somehow indicates to the user the content of the corresponding document 211, 212, 213. Typically, the author of document 200 provides such indications by describing the linked document's contents in the graphic or text element associated with the link. Additional
10 indications of the nature or content of the linked document may be provided by specifying text in an "ALT" tag in the link specification. In some browsers, such as Microsoft Internet Explorer, such text is displayed momentarily on the screen when the user passes an on-screen cursor over the location of the link. Thus, the text may include further description of the contents of the linked document. In addition, many
15 browsers display the URL or file name of a linked document in a status bar whenever an on-screen cursor is passed over a link.

Referring now to Fig. 3, there is shown an example of a screen display of a hypertext document 301 according to the prior art. Document 301 contains text and graphic images; links 302 are indicated and distinguished from normal text by the use
20 of underlining. On-screen cursor 304 can be moved by the user by manipulating pointing device 103. The user activates a link 302 by clicking a button, such as a mouse button, while cursor 304 is positioned over the desired link 302.

In the example of Fig. 3, the user has positioned cursor 304 over link 303. The user is given some indication of the contents of the linked document by the text
25 associated with the link, namely the words "Hewlett-Packard". In addition, a task bar 305 is provided, which displays the URL of the linked document, namely "http://www.hp.com". However, as discussed above, these indications (along with the optional ALT tag, not shown), are insufficient to provide an accurate description of the current contents of the linked document, particularly when such contents may

have changed since the source document 301 was authored. All of the indications shown in Fig. 3 are provided by the author of document 301 only, and are not drawn from or based on the contents of the linked documents themselves. Thus, such indications cannot reflect changes in the linked documents unless the author
5 manually makes corresponding changes to the links in document 301. In addition, none of the indications shown in Fig. 3 provide any graphical representation of the contents of linked documents.

Referring now to Fig. 4, there is shown an example of a screen display of a hypertext document 401 including a link preview 402 according to one embodiment
10 of the present invention. When the user positions cursor 303 over a link 302 and holds cursor 303 in that position for some predetermined length of time, a preview 402 of the linked document is shown. Thus, the user is able to see some portion of the contents of the linked document, and is better able to make a decision as to whether or not to follow the link.

15 In the example of Fig. 4, link preview 402 is shown as a miniature representation of a portion of the linked document, presented in a small window overlapping the display of source document 401. Other arrangements and techniques for displaying link preview 402 may also be used. In addition, the invention may operated in other manners than that described in this example. For example, multiple
20 previews of links could be displayed, each adjacent to the corresponding link. Alternatively, link preview 402 could be displayed only when the user requests such display, for example by clicking a button or activating a command. In yet other embodiments, link previews 402 can be displayed automatically for all or some on-screen links, without requiring the user to express interest in a particular link. Those
25 skilled in the art will recognize that many other implementations of the present invention are possible.

Referring now to Fig. 5, there is shown a block diagram of the functional components of one embodiment of the present invention. In one embodiment, each of the functional components of Fig. 5 is implemented as a software object forming

part of a software application, as described below. In an alternative embodiment, the functional components could be implemented according to some other architecture, as will be recognized by those skilled in the art.

- Link Previewer 501 initiates and coordinates the process of generating and displaying link previews 402. Link Previewer 501 performs the following functions:
- receive messages from Page Viewer 502 that indicate when a link 302, 303 is present in a document;
 - send messages to initiate loading of link previews 402 using Resource Provider 504;
 - respond to user interest or termination of user interest in a link 302, 303 as reported by Page Viewer 502;
 - initiate rendering of link preview 402 using Preview Player 503; and
 - initiate display of link preview 402 in Page Viewer 502 when the user has shown interest in a link 303.

Page Viewer 502 displays source document 401, and delegates a portion of its graphics area to Link Previewer 501 for preview 402 display. Page Viewer 502 also notifies Link Previewer 501 when certain events take place, such as:

- a new document 401 including links 302, 303 is loaded;
- the user indicates interest in a link 303; and
- the user terminates interest in a link 303.

Page Viewer 502 may be implemented using a conventional software system for displaying HTML documents, with appropriate non-conventional enhancements for implementing the features and methods of the present invention.

Preview Player 503 renders a link preview 402 in a portion of Page Viewer's 502 graphics area. Preview Player 503 may also be utilized in link preview resource loading by determining when enough of a link preview resource has been loaded to display the preview 402. In one embodiment, Preview Player 503 is implemented as a portion of Page Viewer 502.

Resource Provider 504 retrieves link preview resources using a location identifier. A location identifier identifies the contents or target of a link. Link preview resources are generally the contents or target themselves, but may be any form of content. In one embodiment, location identifiers are uniform resource locators (URLs), and link preview resources are linked web pages. One skilled in the art will recognize, however, that any content or application may be provided as a resource, and any form of location identifier may be employed, without departing from the spirit or essential characteristics of the present invention.

Location identifiers are provided for Resource Provider 504, which loads the specified content. In one embodiment, Resource Provider 504 performs such functions in a manner that is transparent to the remaining elements of the system. Resource Provider 504 performs such operations in response to signals from Link Previewer 501; it also stops retrieval of resources in response to signals from Link Previewer 501 indicating that a resource is no longer needed or if enough of it has already been loaded. In one embodiment, Resource Provider 504 is capable of loading several resources simultaneously, as needed, though such functionality is not necessary.

Resource Cache 505, which may be implemented in cache 106 of storage device 105, is a repository for stored link preview resources. When a resource is loaded at the request of Link Previewer 501, it is stored in Cache 505 for later retrieval. Resource Cache 505 may store, for example, portions of linked content identified on a document that is currently loaded. In one embodiment, Page Viewer 502 may also use resources stored in Resource Cache 505 for display of the linked document when the user selects a link and thereby requests retrieval of the full document. Thus, Resource Cache 505 aids rendering a link preview more quickly when the user expresses interest, and may also allow linked content to be pre-loaded to improve performance should the user follow the link.

In one embodiment, Resource Cache 505 includes the following methods:

- **cacheResource:** Stores a resource or other content for later retrieval.

- retrieveResource: Returns stored content indexed by location, specified by a location identifier.

Timer 506 is a conventional timer for notifying Link Previewer 501 that a certain amount of time has passed. This information may be used in one embodiment to stop display of a preview 402 after a predetermined length of time has passed. Such length of time may be configurable by the user. In another embodiment, timer 506 may be used to initiate previewing operations only when the user positions the on-screen cursor over a link for a specified length of time. This may be useful in avoiding unwanted preview operations when the user causes the cursor to pass over a link on the way to pointing to another area of the screen. In such a situation, the user has no interest in the link, so that initiating a preview is not desirable.

In one embodiment, Timer 506 includes the following methods:

- startTimer(waittime, expirable): Causes timer to count down until waittime has elapsed.
- expire(): Expires an expirable object passed into startTimer.

In another embodiment, timer 506 may be absent, as it is not necessary to practice the present invention.

The present invention operates in one embodiment by loading resources needed for displaying previews 402 for all links 302, 303 as soon as the source document 401 has been loaded. When the user indicates interest in a particular link 302, the corresponding preview 402 can immediately be displayed if its resources have been loaded. If the loading has not yet been performed or completed when a user expresses interest in a link 303, Link Previewer 501 issues a signal to initiate immediate loading of the resources needed for that link 303. When such loading is complete, the preview 402 is displayed for the user. Preview 402 may be stopped when appropriate, for example when a predefined period of time has elapsed, or when the user indicates that he or she is no longer interested in the link 302.

Referring now to Figs. 6A and 6B, there are shown a timeline diagram and flowchart of a document loading operation according to one embodiment of the

present invention. Document 401 is loaded in response to a request from the user wherein he or she specifies the URL or IP address and location of the desired document. This process is initiated by sending 601 a loadPage(location) instruction to Page Viewer 502. For each link in document 401, steps 602 through 606 are

5 performed. First, Page Viewer 502 notifies 602 Link Previewer 501 of the link using a seenLink(location) call. The seenLink(location) call passes the location identifier to Link Previewer 501 to initiate the preview process, and effectively acts as an entry point for caching linked content for previewing. Link Previewer 501 calls Resource

Cache 505 with an isLinkLoaded(location) call to determine 603 whether or not the

10 link has already been loaded. If the link has not been loaded, and if in 604 it is determined that loading of the link has not already begun, Link Previewer 501 calls Resource Provider 504 with a loadLink(location) command to initiate loading. Resource Provider 504 initiates loading, and sends a cacheResource(location, resource) command 606 to Resource Cache 505 in order to cache to loaded link for

15 future use.

Operations shown in Figs. 6A and 6B may proceed in parallel, so that some links may continue to load while other operations take place. Thus, loading may still be in process when the user expresses an interest in a link, as described below in connection with Figs. 7A and 7B.

20 Referring now to Figs. 7A and 7B, there are shown a timeline diagram and flowchart of a link previewing operation according to one embodiment of the present invention. The method of Figs. 7A and 7B allow for concurrent loading operations as described above, with priority given to any link in which the user expresses interest. In other words, the system according to one embodiment forces immediate loading of

25 preview data for a link whenever the user holds the cursor over that link or otherwise expresses interest.

A user expresses interest 700 in a link generally by positioning an on-screen cursor over the link and holding that position for some predetermined length of time, such as for example one-half second, though any length of time may be used.

Typically, Page Viewer 502 receives a signal indicating that the cursor has been so positioned, as is known in the art for the display of ToolTips or similar descriptive information concerning the link.

Page Viewer 502 sends an interest(location) command 701 to Link Previewer 501, indicating that a preview should be displayed for the specified location (i.e., link). Link Previewer 501 then sends an isLinkLoaded(location) command 702 to Resource Cache 505 to determine whether preview data for the specified link has already been loaded. If not, Link Previewer 501 sends a forceLoad(location) command 703 to Resource Provider 504 forcing immediate loading of the relevant preview data. This ensures that priority will be given to the display of the particular link the user has indicated, without wasting undue time loading resources for other links. Loading of other links can resume when it is possible to do so without compromising the speed or efficiency of loading the desired link.

Link Previewer 501 sends a getPreviewArea(size) command 704 to Page Viewer 502. The getPreviewArea(size) command provides a GraphicsArea parameter to Preview Player 503, so that Preview Player 503 can display the preview in the appropriate location. Link Previewer 501 then sends a playPreview(size, GraphicsArea) command 705 to Preview Player 503, which begins to display the preview in the specified location on the display screen. This display may show a miniature representation of a portion of the linked document, as shown in the example of Fig. 1.

In one embodiment, Link Previewer 501 also sends a startTimer(duration) command 706 to Timer 506, specifying a duration for display of the preview. The purpose of this step is to provide a mechanism for stopping the display of the preview after some predetermined length of time, so that the preview is not obtrusive to the user. This length of time may be user-configurable, and there may also be alternative mechanisms for the user to request that the preview display be stopped. In addition, if the user moves the on-screen cursor so that it is no longer positioned

over the link being previewed, the preview is stopped, as will be described below in connection with Figs. 8A and 8B.

Once Timer 506 determines that the duration of time has expired, it sends an expired() command 707 to Link Previewer 501, which in turn sends a stopPreview()
 5 command 708 to Preview Player 503. Preview Player 503 stops displaying the preview. Link Previewer 501 also sends a refresh() command 709 to Page Viewer 502 which refreshes the area of the display that was occupied by the preview.

Referring now to Figs. 8A and 8B, there are shown a timeline diagram and flowchart of a preview stopping operation according to one embodiment of the
 10 present invention. These Figures show operation of the present invention when the user indicates that he or she is no longer interested in a link.

The user indicates 800 that he or she is no longer interested in a link, typically by moving the on-screen cursor so that it is no longer positioned over the link. Page Viewer 502 detects this, and sends a notInterested(location) command 801 to Link
 15 Previewer 501 to signal the event. Link Previewer 501 sends stop(location) commands 802, 803, 804, 805 to Preview Player 503, Resource Provider 504, Resource Cache 505, and Timer 506, respectively. This serves to cease display of the link preview by notifying all components that the user is no longer interested in the link.

In one embodiment, loading of link preview resources continues concurrently
 20 with the steps of Figs 7A, 7B, 8A, and 8B. Thus, previews of linked documents are continuously loaded until loading is complete, so as to minimize display of a preview when the user expresses interest in a link.

From the above description, it will be apparent that the invention disclosed herein provides a novel and advantageous system and method of displaying a
 25 preview of a linked document before the link is activated. The foregoing discussion discloses and describes merely exemplary methods and embodiments of the present invention. As will be understood by those familiar with the art, the invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. Accordingly, the disclosure of the present invention is

intended to be illustrative, but not limiting, of the scope of the invention, which is set forth in the following claims.

We Claim:

1. In a system for displaying hypertext documents, a computer-implemented method of displaying a preview of a linked document, comprising:

- a) retrieving a source document containing at least one link to a linked document;
- b) receiving a selection identifying one of the links;
- c) retrieving preview data representing at least a portion of the linked document corresponding to the identified link; and
- d) displaying the retrieved preview data.

2. The computer-implemented method of claim 1, wherein c) comprises:

- c.1) retrieving at least a portion of the linked document; and
- c.2) generating preview data from the retrieved portion of the linked document.

3. The computer-implemented method of claim 1, further comprising:

- e) stopping display of the retrieved preview data after a predetermined time period.

4. The computer-implemented method of claim 1, wherein the at least one link has an on-screen position, and wherein b) comprises detecting that a position of an on-screen cursor corresponds to the position of the link.

5. The computer-implemented method of claim 4, further comprising:

- e) stopping display of the retrieved preview data responsive to the position of the on-screen cursor no longer corresponding to the position of the link.

6. The computer-implemented method of claim 1, further comprising:

- e) stopping display of the retrieved preview data responsive to the position of the on-screen cursor no longer corresponding to the position of the link.

5 7. The computer-implemented method of claim 1, further comprising:
subsequent to step a), for each link in the source document:

- a.1) retrieving preview data representing at least a portion of the linked document corresponding to the link; and
- a.2) caching the retrieved preview data in a storage cache;

10 and wherein step c) comprises retrieving preview data from the storage cache.

8. The computer-implemented method of claim 1, further comprising:
subsequent to step a), for each link in the source document:

- a.1) commencing retrieval of preview data representing at least a portion of the linked document corresponding to the link; and
- a.2) commencing caching of the retrieved preview data in a storage cache;

and wherein step c) comprises:

- c.1) responsive to retrieval and caching of the preview data corresponding to the identified link being complete, retrieving preview data from the storage cache; and
- c.2) responsive to retrieval and caching of the preview data corresponding to the identified link not being complete, forcing retrieval of preview data corresponding to the identified link.

9. The computer-implemented method of claim 1, wherein d) comprises displaying the retrieved preview data in an overlay window adjacent to the identified link.

10. The computer-implemented method of claim 1, wherein d) comprises displaying a reduced-size representation of a portion of the linked document in an overlay window adjacent to the identified link.

11. A system for displaying a preview of a linked document in a hypertext document display application, comprising:

10 a retrieval module for retrieving a source document containing at least one link to a linked document;
an input device coupled to the retrieval module, for receiving a selection identifying one of the links;
a preview generating module, coupled to the retrieval module, for retrieving preview data representing at least a portion of the linked document
15 corresponding to the identified link; and
a viewer, coupled to the preview generating module, for displaying the retrieved preview data.

12. The system of claim 11, wherein the preview generating module comprises:

20 a resource provider for retrieving at least a portion of the linked document;
and
a link previewer, coupled to the resource provider, for generating preview data from the retrieved portion of the linked document.

13. The system of claim 11, wherein the at least one link has an on-screen position, and wherein the input device detects that a position of an on-screen cursor
25 corresponds to the position of the link.

14. The system of claim 13, wherein the preview generating module stops display of the retrieved preview data responsive to the position of the on-screen cursor no longer corresponding to the position of the link.

15. The system of claim 11, further comprising:

a resource cache coupled to the retrieval module for, for each link in the source document, retrieving preview data representing at least a portion of the linked document corresponding to the link, and caching the retrieved preview data in a storage cache;

and wherein the preview generating module retrieves preview data from the storage cache.

16. The system of claim 11, wherein the viewer displays the retrieved preview data in an overlay window adjacent to the identified link.

17. The system of claim 11, wherein the viewer displays a reduced-size representation of a portion of the linked document in an overlay window adjacent to the identified link.

18. A computer program product comprising a computer-usable medium having computer readable code embodied therein for displaying a preview of a linked document in a system for displaying hypertext documents, the computer program product comprising:

computer-readable program code devices configured to cause a computer to retrieve a source document containing at least one link to a linked document;

computer-readable program code devices configured to cause a computer to receive a selection identifying one of the links;

5 computer-readable program code devices configured to cause a computer to
retrieve preview data representing at least a portion of the linked
document corresponding to the identified link; and
computer-readable program code devices configured to cause a computer to
display the retrieved preview data.

19. The computer program product of claim 18, wherein the computer-
readable program code devices configured to cause a computer to retrieve preview
data comprise:

10 computer-readable program code devices configured to cause a computer to
retrieve at least a portion of the linked document; and
computer-readable program code devices configured to cause a computer to
generate preview data from the retrieved portion of the linked
document.

15 20. The computer program product of claim 18, further comprising:
computer-readable program code devices configured to cause a computer to
stop display of the retrieved preview data after a predetermined
time period.

21. The computer program product of claim 18, wherein the at least one link
has an on-screen position, and wherein the computer-readable program code devices
20 configured to cause a computer to receive a selection identifying one of the links
comprise computer-readable program code devices configured to cause a computer
to detect that a position of an on-screen cursor corresponds to the position of the link.

22. The computer program product of claim 21, further comprising:
computer-readable program code devices configured to cause a computer to
25 stop display of the retrieved preview data responsive to the position

of the on-screen cursor no longer corresponding to the position of the link.

23. The computer program product of claim 18, further comprising:

computer-readable program code devices configured to cause a computer to
5 stop display of the retrieved preview data responsive to the position
of the on-screen cursor no longer corresponding to the position of
the link.

24. The computer program product of claim 18, further comprising:

computer-readable program code devices configured to cause a computer to,
10 for each link in the source document:
retrieve preview data representing at least a portion of the linked
document corresponding to the link; and
cache the retrieved preview data in a storage cache;

and wherein the computer-readable program code devices configured to cause
15 a computer to retrieve preview data comprise computer-readable program code
devices configured to cause a computer to retrieve preview data from the storage
cache.

25. The computer program product of claim 18, further comprising:

computer-readable program code devices configured to cause a computer to,
20 for each link in the source document:
commence retrieval of preview data representing at least a portion
of the linked document corresponding to the link; and
commence caching of the retrieved preview data in a storage cache;

and wherein the computer-readable program code devices configured to cause
25 a computer to retrieve preview data comprise:

computer-readable program code devices configured to cause a computer to, responsive to retrieval and caching of the preview data corresponding to the identified link being complete, retrieve preview data from the storage cache; and

5 computer-readable program code devices configured to cause a computer to, responsive to retrieval and caching of the preview data corresponding to the identified link not being complete, force retrieval of preview data corresponding to the identified link.

26. The computer program product of claim 18, wherein the computer-
10 readable program code devices configured to cause a computer to display the retrieved preview data comprise computer-readable program code devices configured to cause a computer to display the retrieved preview data in an overlay window adjacent to the identified link.

27. The computer-implemented method of claim 18, wherein the computer-
15 readable program code devices configured to cause a computer to display the retrieved preview data comprise computer-readable program code devices configured to cause a computer to display a reduced-size representation of a portion of the linked document in an overlay window adjacent to the identified link.

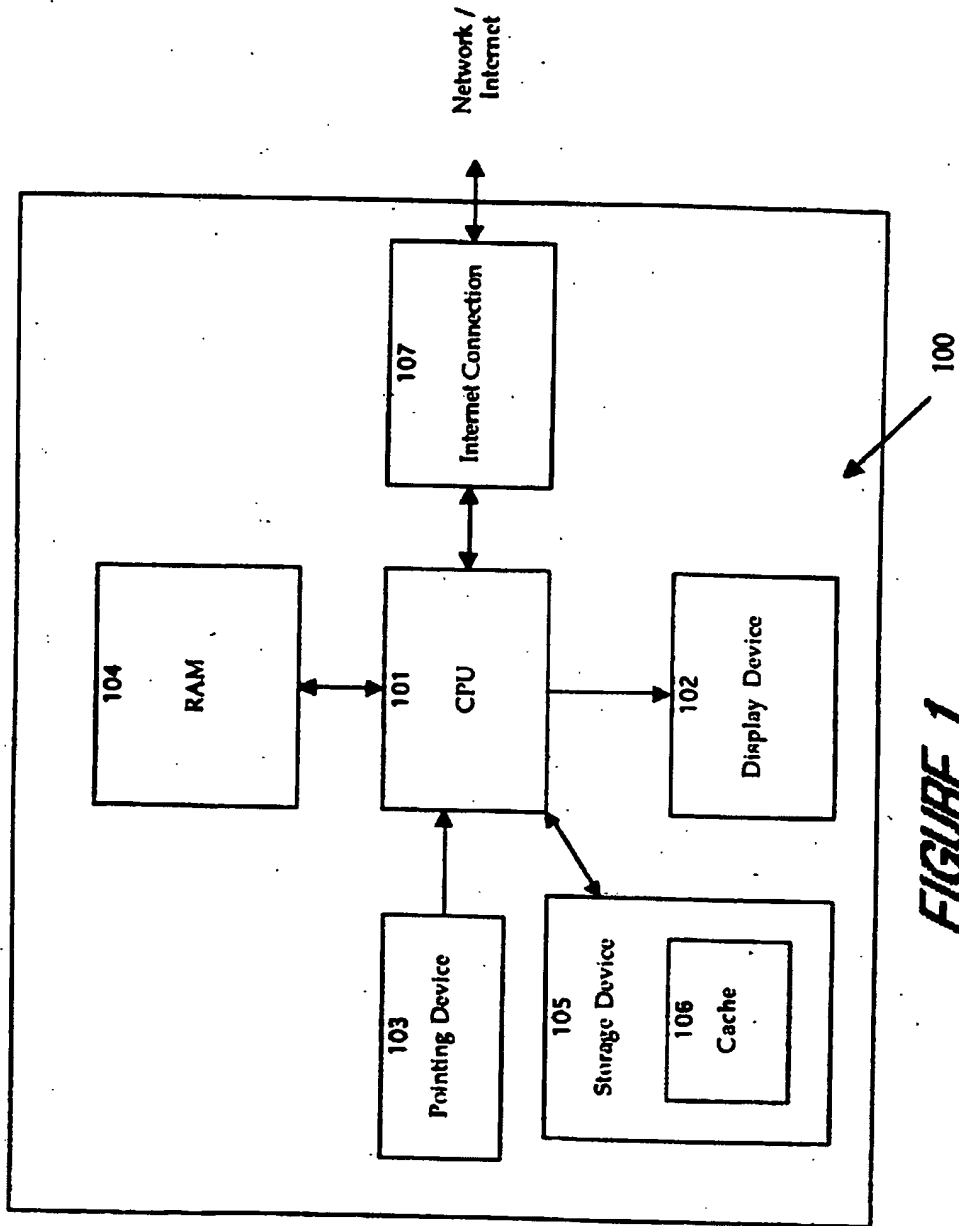


FIGURE 1

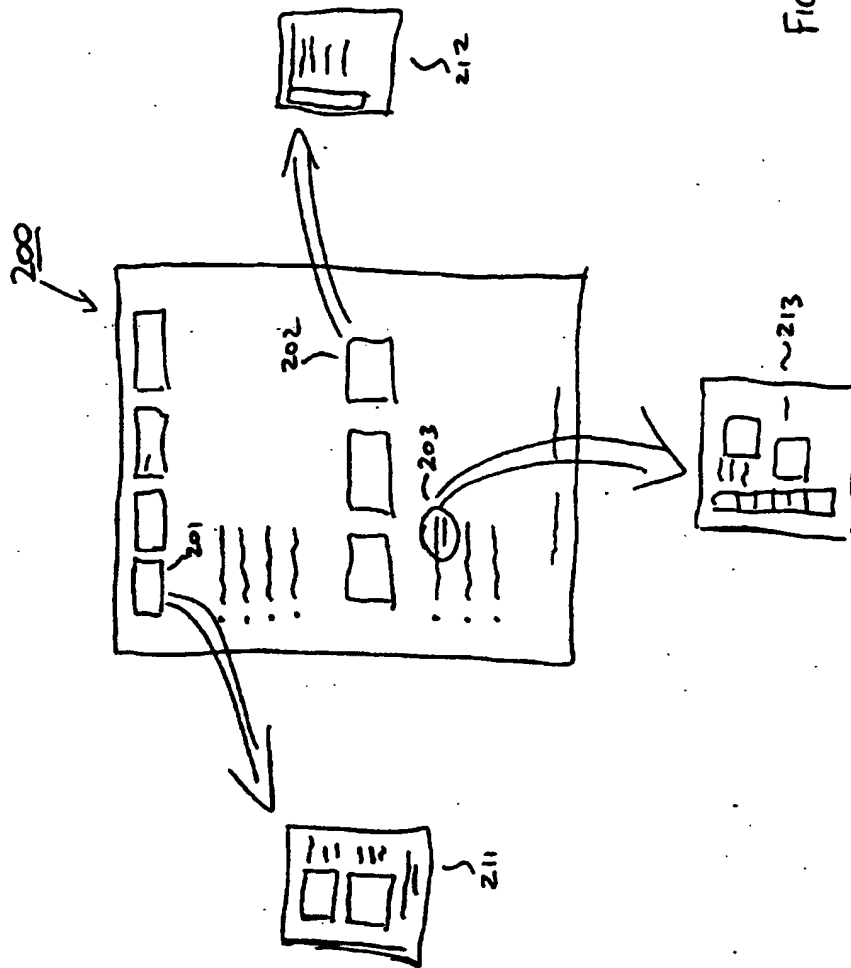


FIG 2

File Edit View Go Bookmarks Options Directory Window [2] Connected

NetScape: Corel Partners


Back Forward Home Edit Reload Images Print Find Stop

NetSite: http://www.corel.com/partners_developers/partners.htm

What's New? What's Cool? Destinations Net Search People Software

COREL Home Products Support Shop Contact Us News Events Search

Corel Partners



Technology Partners

- Dragon Systems, Inc.
- Amplitude Software
- Bluekila International Pvt. Ltd.
- Borland
- CNLC
- Horlman + Associates
- NSO
- KMT Software
- NetScape
- Novell
- Sterish Software
- Software Development Corporation
- Santo
- Intel Corporation

OEM Partners

- Epson
- Hewlett-Packard
- Packard Bell
- UNIQ
- Vides

Learning Partners

- Approved Curriculum Vendors
- Conferences and Seminars
- Magazines, Books, Newsletter
- SchoolNet
- Training Partners

301

302

303

304

305

7/1/00 <http://www.hp.com>

Fig. 3

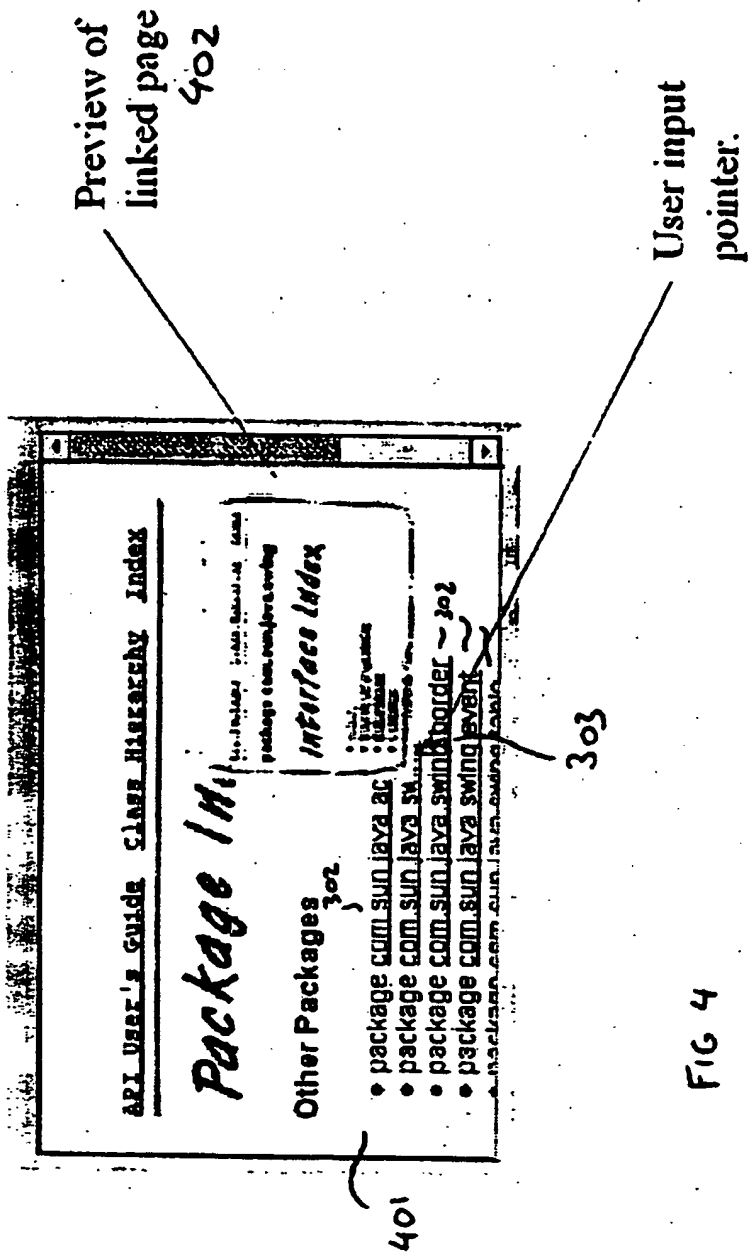


FIG 4

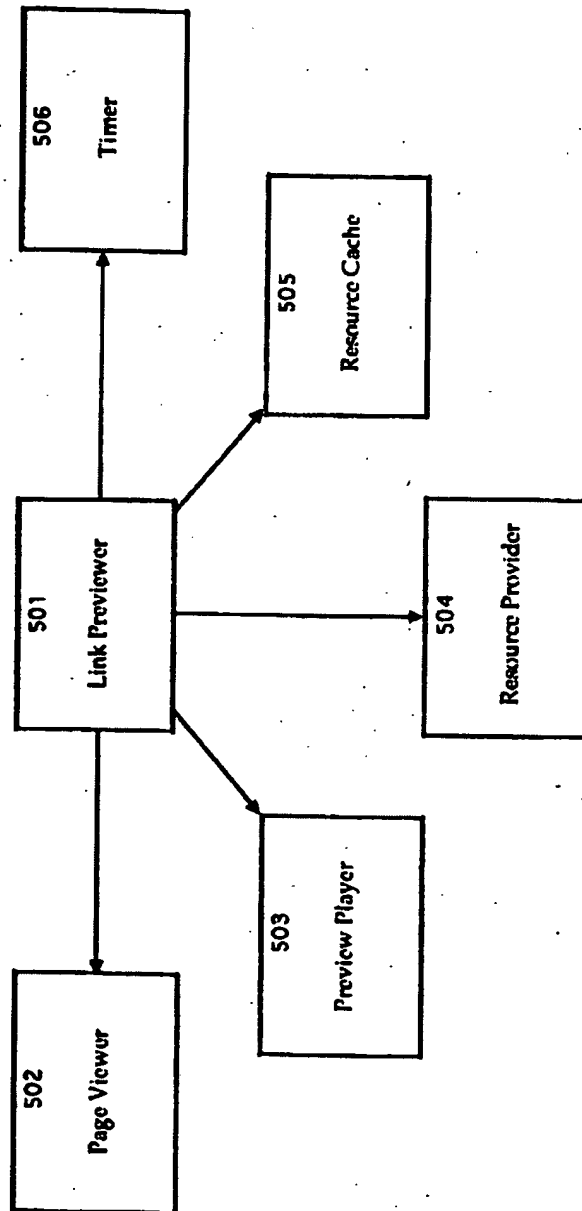


FIGURE 5

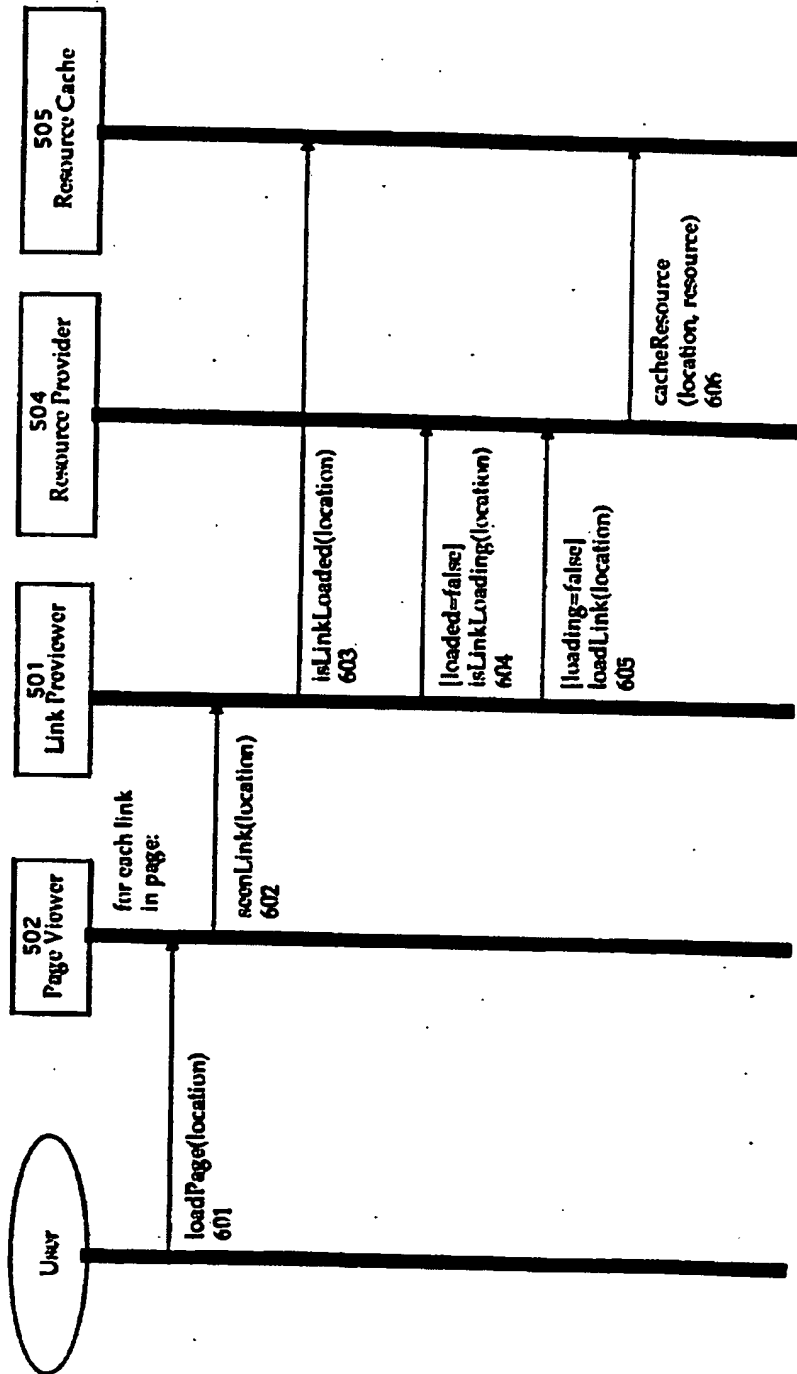
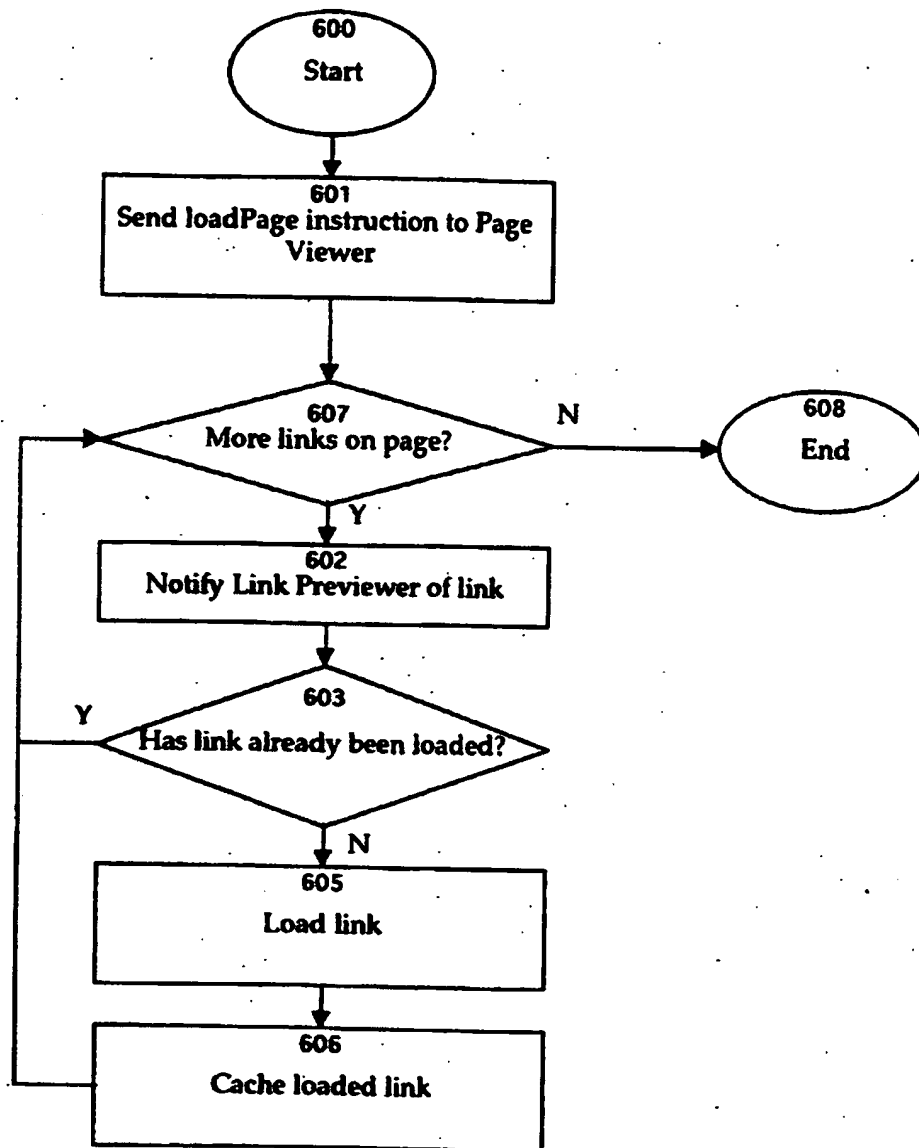


FIGURE 6A

**FIGURE 6B**

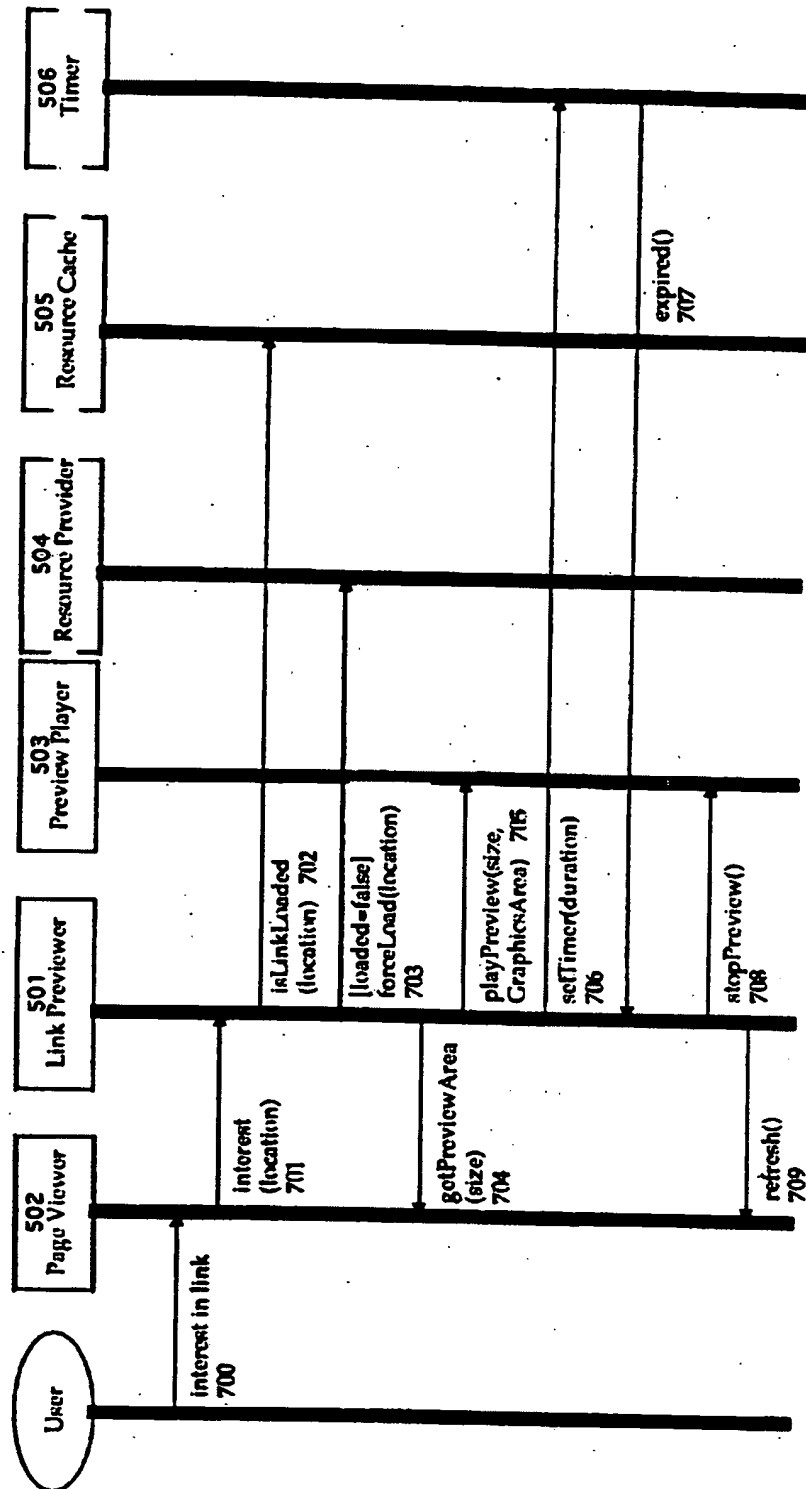
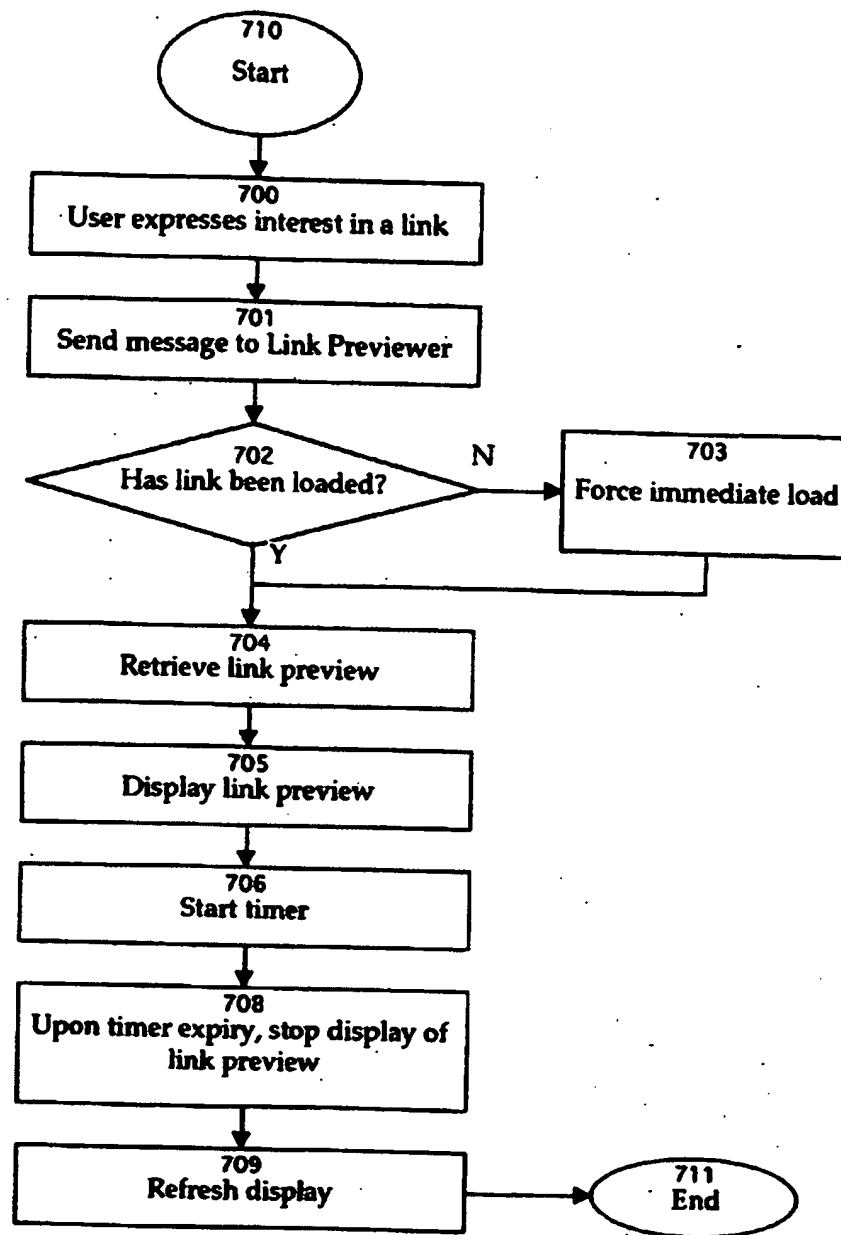


FIGURE 7A

**FIGURE 7B**

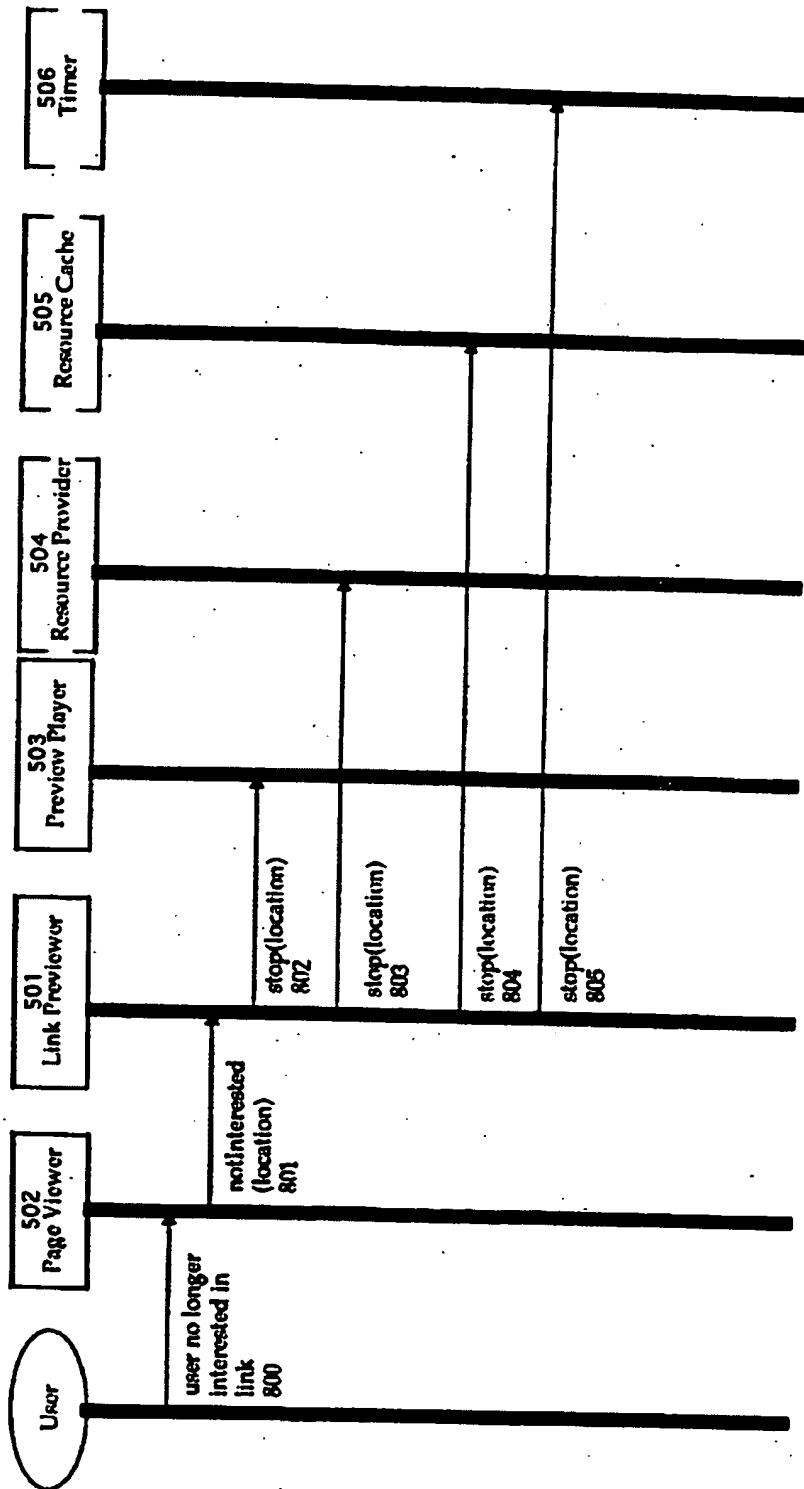


FIGURE 8A

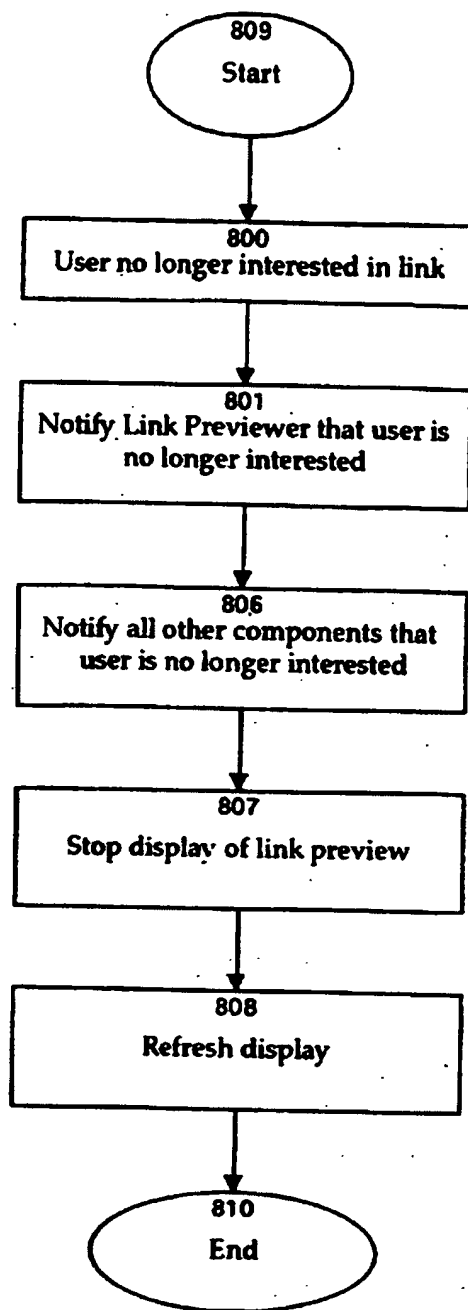


FIGURE 8B

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.